Práctico 3. Clase 4 de Análisis y Curación

Andrés Vázquez y Sergio Buzzi

Inciso 1

- Elija un dataset clasificado de su preferencia y area (domain expertise), aplique un metodo de clustering y/o mixtura de Gaussianas en el mismo.

Datos

Se tiene información sobre 46 variables, para los 479 barrios de la Ciudad de Córdoba, originaria de la Encuesta Provincial de hogares de la Provincia de Córdoba 2008.

```
#load("base.RData")
# En base se tiene la información de 46 variables
#save(base, file="datosbarrios.RData")
#dim(base)
load("datosbarrios.RData")
```

```
attach(base)
educ=base[,1:11]
leer=base[,12:13]
salud=base[,14:15]
empleo=base[,16:18]
nbi=base[,20:24]
privacion=base[,25:28]
vivienda=base[,29:33]
habitantes=base[,34:41]
std=base[,42:46]

#Para que tenga sentido el análisis se toman algunos ratios por cuestiones conceptuales,
# por ejemplo la cantidad de personas empleadas en un barrio esta influenciada por las personas en
# condición de trabajar (14 años o mas)
educstd=educ/jefes
leerstd=leer/pob3omas
saludstd=salud/poblacion
empleostd=empleo/pob14omas
nbistd=nbi/poblacion
privacionstd=privacion/poblacion
viviendastd=vivienda/hogares
habitantesstd=habitantes/hogares

basestd=cbind(educstd,leerstd,saludstd,empleostd,nbistd,privacionstd,viviendastd,habitantesstd)
sub=subset(cbind(basestd,poblacion),poblacion>=2000)
semibase=sub[,-ncol(sub)]

# se saca del listado sacar del listado: inicial, sabeleer, cobertura, ocupados, inactivos, sinprivacio
datos=semibase[,-c(1,12,14,16,18,24,28,33)]
dat=datos
```

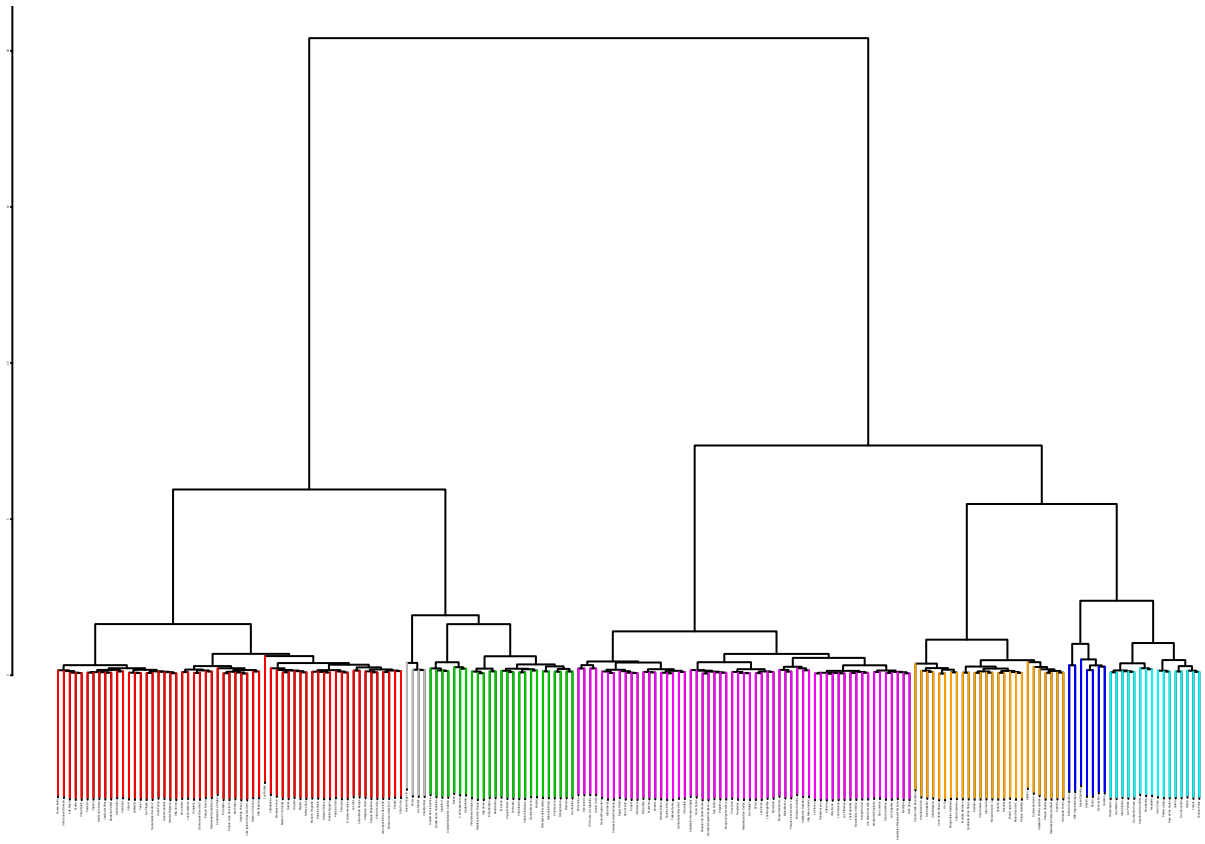
Aplicación de clustering

Un alogritmo de clustering jerárquico

```
#cluster jerárquico
hc = hclust(dist(dat), method = 'ward')
```

```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```
par(mar=c(0,5,0,0), cex=0.08)
y=cutree(hc, 7)
library(sparcl)
ColorDendrogram(hc, y = y, labels = names(y), branchlength = 4)#plot52
```



Clustering por kmeans con un k arbitrario (k=3)

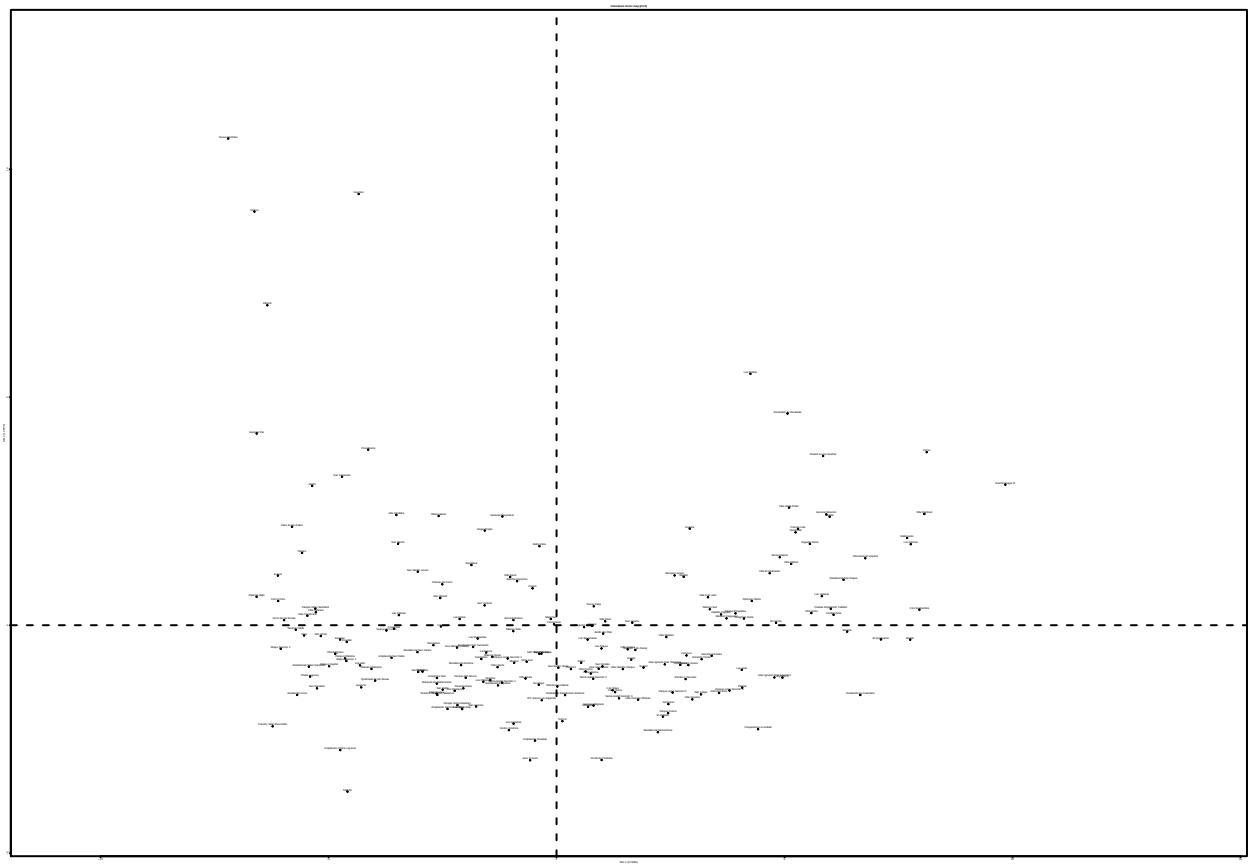
```
library(fpc)
fit=kmeansruns(dat,krange=3,criterion="ch")
```

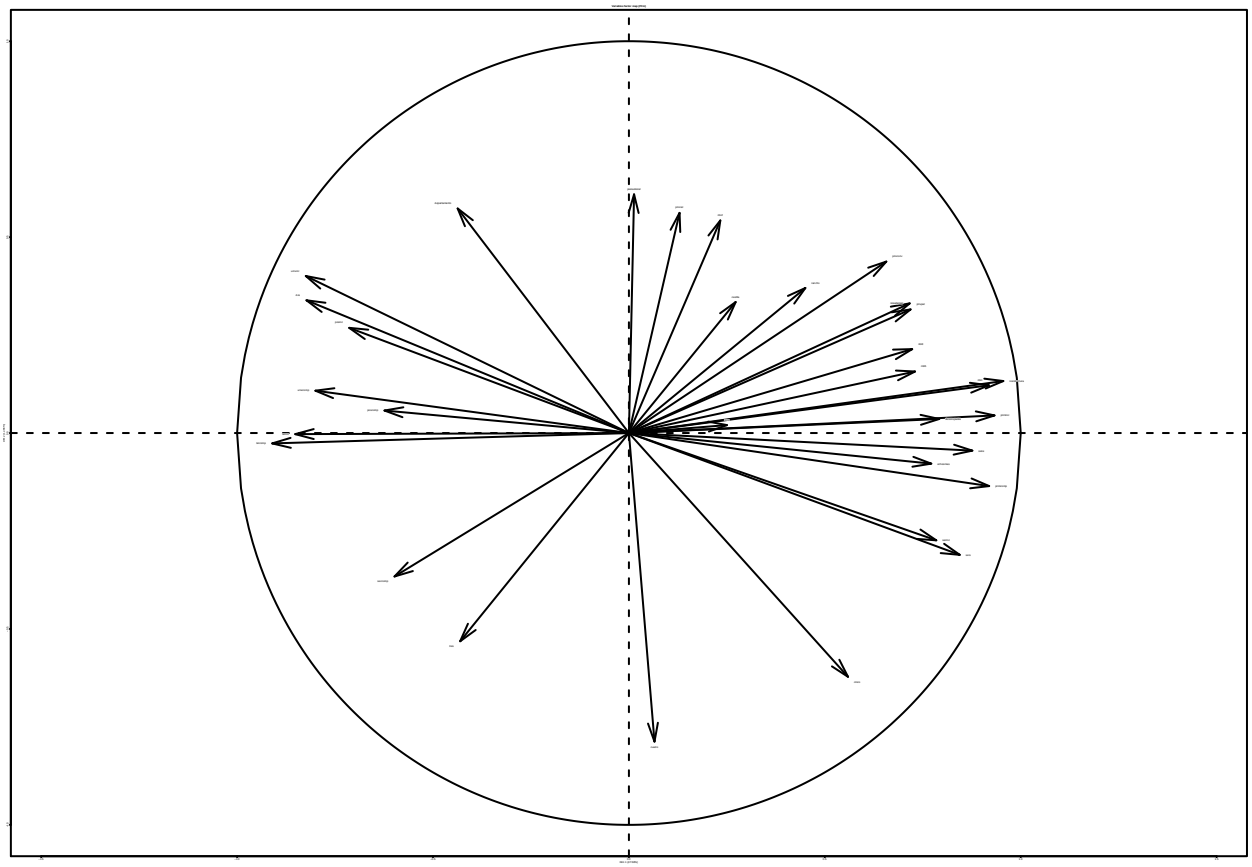
En muchos casos se recomienda armar los grupos en base a los componentes principales mas importantes. Del siguiente modo se puede aplicar componentes principales:

```
library(FactoMineR)
```

```
## Warning: package 'FactoMineR' was built under R version 3.5.3
```

```
par(cex=0.05)
result <- PCA(dat)
```





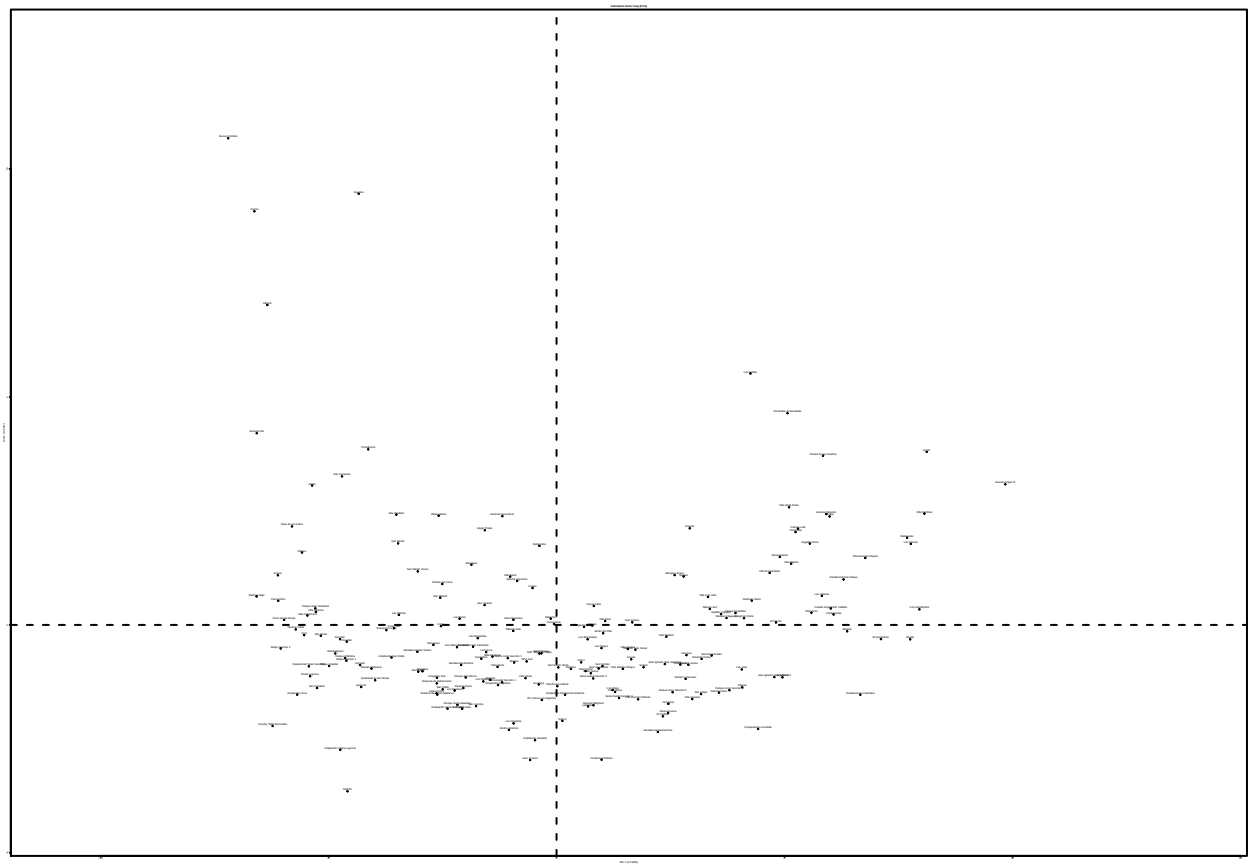
```
summary(result)
```

```
##
## Call:
## PCA(X = dat)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6
## Variance              15.226   4.051   2.582   2.150   1.137   1.015
## % of var.             47.582  12.660   8.068   6.718   3.553   3.172
## Cumulative % of var.  47.582  60.242  68.310  75.028  78.581  81.753
##                        Dim.7   Dim.8   Dim.9  Dim.10  Dim.11  Dim.12
## Variance               0.772   0.701   0.595   0.543   0.482   0.391
## % of var.              2.412   2.190   1.861   1.696   1.507   1.222
## Cumulative % of var.  84.165  86.355  88.216  89.911  91.419  92.641
##                       Dim.13  Dim.14  Dim.15  Dim.16  Dim.17  Dim.18
## Variance               0.321   0.297   0.260   0.233   0.190   0.184
## % of var.              1.002   0.927   0.811   0.729   0.593   0.576
## Cumulative % of var.  93.643  94.570  95.381  96.110  96.703  97.279
##                       Dim.19  Dim.20  Dim.21  Dim.22  Dim.23  Dim.24
## Variance               0.145   0.124   0.117   0.101   0.081   0.073
## % of var.              0.453   0.387   0.366   0.316   0.254   0.229
## Cumulative % of var.  97.731  98.118  98.484  98.800  99.054  99.283
##                       Dim.25  Dim.26  Dim.27  Dim.28  Dim.29  Dim.30
```

```
## Variance               0.052   0.046   0.038   0.037   0.029   0.018
## % of var.              0.163   0.144   0.118   0.115   0.089   0.056
## Cumulative % of var.  99.446  99.590  99.709  99.824  99.913  99.970
##                       Dim.31  Dim.32
## Variance               0.010   0.000
## % of var.              0.030   0.000
## Cumulative % of var. 100.000 100.000
##
## Individuals (the 10 first)
##                      Dist    Dim.1    ctr   cos2    Dim.2    ctr   cos2
## 1 de Mayo         |  2.487 |  1.231  0.051  0.245 | -1.427  0.259  0.329 |
## 2 de Septiembre   |  9.444 |  7.960  2.145  0.710 |  0.341  0.015  0.001 |
## Acosta            |  2.957 |  1.639  0.091  0.307 | -0.754  0.072  0.065 |
## Alberdi           |  9.956 | -6.340  1.361  0.406 |  7.016  6.264  0.497 |
## Alta Córdoba      |  4.682 | -3.510  0.417  0.562 |  2.418  0.744  0.267 |
## Altamira          |  3.243 | -1.450  0.071  0.200 | -1.204  0.185  0.138 |
## Alto Alberdi      |  2.885 | -2.549  0.220  0.781 |  0.597  0.045  0.043 |
## Alto Verde        |  5.570 | -5.169  0.904  0.861 | -0.238  0.007  0.002 |
## Ameghino Norte    |  5.251 |  4.112  0.572  0.613 |  0.148  0.003  0.001 |
## Ameghino Sud      |  3.801 | -2.622  0.233  0.476 | -1.157  0.170  0.093 |
##                    Dim.3    ctr   cos2
## 1 de Mayo         -0.803  0.129  0.104 |
## 2 de Septiembre    0.052  0.001  0.000 |
## Acosta            -1.555  0.483  0.276 |
## Alberdi           -1.811  0.655  0.033 |
## Alta Córdoba       0.652  0.085  0.019 |
## Altamira          -0.791  0.125  0.060 |
## Alto Alberdi      -0.451  0.041  0.024 |
## Alto Verde         1.072  0.229  0.037 |
## Ameghino Norte    -0.375  0.028  0.005 |
## Ameghino Sud      -0.518  0.054  0.019 |
##
## Variables (the 10 first)
##                     Dim.1    ctr   cos2    Dim.2    ctr   cos2    Dim.3
## priminc           |  0.933  5.719  0.871 |  0.045  0.050  0.002 | -0.099
## primcomp          |  0.919  5.552  0.845 | -0.135  0.452  0.018 | -0.224
## secinc            |  0.785  4.042  0.616 | -0.274  1.851  0.075 | -0.290
## seccomp           | -0.598  2.351  0.358 | -0.366  3.309  0.134 | -0.118
## terinc            | -0.852  4.765  0.725 | -0.003  0.000  0.000 | -0.043
## tercomp           | -0.910  5.444  0.829 | -0.026  0.017  0.001 |  0.225
## univinc           | -0.825  4.467  0.680 |  0.401  3.966  0.161 | -0.038
## univcomp          | -0.800  4.206  0.640 |  0.108  0.288  0.012 |  0.389
## posinc            | -0.714  3.348  0.510 |  0.268  1.780  0.072 |  0.295
## poscomp           | -0.624  2.557  0.389 |  0.057  0.081  0.003 |  0.471
##                     ctr   cos2
## priminc           0.379  0.010 |
## primcomp          1.938  0.050 |
## secinc            3.256  0.084 |
## seccomp           0.536  0.014 |
## terinc            0.070  0.002 |
## tercomp           1.965  0.051 |
## univinc           0.056  0.001 |
## univcomp          5.867  0.151 |
## posinc            3.378  0.087 |
```

```
## poscomp          8.584  0.222 |
```

```
plot(result) # gráficos varios entre ellos el biplot
```



```
res=result$ind$coord[]
#primeros dos componentes
x1 = res[,1]
x2 = res[,2]
```

y luego correr kmeans, sobre los primeros componentes principales, por ejemplo los dos primeros:

```
fit=kmeansruns(cbind(x1,x2),krange=3,criterion="ch")
```

Inciso 2

- Investigue los resultados en el meta parametro K numero de cumulos e investigue posibles procesos de seleccion del mismo.

Se puede aplicar kmeans para diversos k usando el argumento krange de la función kmeansruns, por ejemplo para $k = 1, ...10$. Para seleccionar k, se podría acudir alguna función de la suma de cuadrados dentro de los clusters y entre clusters. En la misma función kmeansruns estan implementados algunos criterios de selección de k (average silhouette width y Calinski-Harabasz). Por ejemplo aquí se implementa la selección del k optimo entre $k = 1, ...10$, por el método Calinski-Harabasz:

```
fit = kmeansruns(dat,krange=1:10,criterion="ch")
```

El algoritmo indica que el optimo es k=2.

Inciso 3

- Comente la influencia de la normalización de los datos en los resultados del clustering.

```
fit =kmeansruns(dat,krange=1:10,criterion="ch", scaledata=TRUE)
```

Al trabajar con los datos estandarizados se encuentran dos clusters. La estandarización no modificó demasiado el análisis. Esto puede deberse a que las transformaciónes previar realizadas sobre las variables generan cierto grado de estandarización.

Finalmente, se grafican los clusters y se ven cuantos barrios hay en cada uno de ellos

```
library(cluster)
#fit1 <- kmeans(dat, 2, nstart=100, scale=TRUE)
par(mar=c(5,5,1,1), cex=0.4)
clusplot(dat, fit$cluster, color=TRUE, shade=TRUE,
         labels=2, lines=0, cex=0.8)
```

```r
#par(opar)
table(fit$cluster)
```

```
##
##   1   2
## 107  87
```